

Filtrage ensembliste multi-hypothèse sur des données intervalles

Application à la localisation

Fahed ABDALLAH, Amadou GNING, Philippe BONNIFAIT

HEUDIASYC, UMR CNRS 6599
 Université de Technologie de Compiègne
 B.P. 20529, 60205 Compiègne Cedex, FRANCE
 fahed.abdallah@hds.utc.fr

Résumé –

Le filtrage particulaire est une méthode utilisée pour l'estimation d'état des systèmes dynamiques. L'un des inconvénients majeurs de ce type de filtrage est la nécessité d'utiliser un grand nombre de particules pour explorer les régions de forte vraisemblance et éviter ainsi une divergence éventuelle du filtre. Dans cet article, nous proposons un algorithme hybride basé simultanément sur le filtrage particulaire et l'analyse par intervalles. La méthode proposée permet de réduire significativement le nombre de particules par rapport au filtrage particulaire classique et de répondre ainsi aux applications temps réel. Des résultats sur un problème de localisation d'un mobile montrent l'utilité et l'efficacité de la méthode proposée.

Abstract –

Particle Filtering is used in nonlinear state estimation for dynamic systems. For some applications, PF are not adapted for real-time implementations. In this paper we propose a new method based simultaneously on particle filters and interval analysis and which is well adapted for real time applications. Intervals will allow to describe regions with high likelihood by a small number of box particles instead of a large number of particles like in the case of PF. Experiments on real data for global vehicle localization show the usefulness and the efficiency of the proposed approach.

1 Introduction

Les méthodes bayésiennes fournissent un cadre attrayant pour les problèmes d'estimation dynamique de l'état d'un système [3]. L'approche bayésienne consiste à construire la fonction densité de probabilité (FDP) du vecteur d'état en se basant sur une formulation probabiliste des informations disponibles. Le filtrage particulaire est une méthode séquentielle qui permet de mettre en œuvre le filtrage bayésien en utilisant les techniques dites de Monte Carlo. Le principe de Monte Carlo consiste à utiliser un échantillon discret (ou ensemble de particules) obtenu par simulation pour approximer une distribution de probabilité que l'on ne sait pas calculer analytiquement. L'efficacité et la pertinence de la méthode du filtre particulaire dépendent, la plupart du temps, du nombre de particules utilisé dans l'approximation de la FDP.

Si l'imprécision dans l'information disponible est importante, le nombre de particules doit être très grand afin d'obtenir de bonnes performances et d'éviter une éventuelle divergence du filtre. Ceci peut induire des problèmes de complexité pour une mise en œuvre en temps réel de la méthode. Plusieurs travaux récents essayent de combiner des approches afin de surmonter ces imperfections (voire par exemple [6] et les références incluses). D'autres approches utilisent des approches statistiques pour augmenter l'efficacité des filtres particuliers en adaptant la taille de l'échantillon pendant le processus d'estimation en fonction de la précision de l'information disponible [4].

Avec l'hypothèse qu'on connaît seulement une borne max-

imale des erreurs sur les données disponibles, il apparaît judicieux de modéliser ces données en utilisant des intervalles. Les méthodes qui utilisent une telle modélisation rentrent dans le cadre des méthodes à erreurs bornées. L'estimation paramétrique à erreurs bornées représente une direction de recherche relativement nouvelle dans quelques applications du traitement du signal. Dans le domaine de la commande, étroitement lié, il y a eu des travaux montrant l'utilité et l'efficacité de telles approches. Dans cet article, nous proposons un algorithme hybride de filtre particulaire en traitant des données intervalles et en employant des techniques et outils d'analyse par intervalle et de satisfaction de contraintes [7]. Dans le filtrage particulaire classique, les particules représentent des états ponctuels dont les poids associés dépendent des vraisemblances définies à partir d'un modèle statistique construit sur l'erreur d'observation. Dans le filtre à pavés (également appelés boîtes), les particules sont des pavés et les poids associés sont définis par un modèle borné de l'erreur d'observation.

2 Filtrage particulaire

Le modèle d'évolution peut être décrit par le système :

$$\begin{cases} x_{k+1} = f(x_k, u_k, v_k) \\ y_k = g(x_k, w_k) \end{cases} \quad (1)$$

où $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ est une fonction, éventuellement non linéaire, définissant la relation entre l'état à l'instant $k + 1$, l'état l'instant k , l'entrée u_k et une séquence d'un bruit

$v_k, k \in \mathbb{N}$. Les dimensions de l'état, de l'entrée et du bruit sont respectivement n_x, n_u et n_v . La fonction $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$ est une fonction, éventuellement non linéaire, définissant la relation entre l'état et la mesure à l'instant k , $w_k, k \in \mathbb{N}$, modélise le bruit considéré sur la mesure. n_y, n_w sont les dimensions respectives de la mesure et de son bruit. Les états et les mesures jusqu'à l'instant k sont représentés par $X_k = \{x_i, i = 1, \dots, k\}$ et $Y_k = \{y_i, i = 1, \dots, k\}$, respectivement.

D'un point de vue Bayésien, en supposant qu'on dispose de Y_k , tout ce qu'on désire savoir sur l'état à l'instant k est donné par la FDP $p(X_k|Y_k)$, nommée densité a posteriori. Ainsi, l'a posteriori constitue la solution complète du problème d'estimation séquentielle de l'état. Dans les applications réelles, on est souvent ramené à estimer une marginale de l'a posteriori, $p(x_k|Y_k)$. Cette fonction marginale représente la croyance sur l'état x_k à l'instant k , prenant plusieurs valeurs, en supposant qu'on dispose des mesures $Y_k = \{y_i, i = 1, \dots, k\}$. Connaissant $p(x_k|Y_k)$, on peut calculer aisément différentes estimées de l'état du système comme la moyenne, le mode ou des intervalles de confiance.

Les filtres particulaires sont des méthodes qui permettent d'approximer la densité de probabilité $p(x_k|Y_k)$ via un nuage de N particules avec des masses, ou poids associés. Ainsi, une densité de probabilité, éventuellement continue, est approximée par une loi discrète. Le scénario d'un filtre particulaire est comme suit. Initialement, toutes les particules possèdent le même poids. Pour estimer l'état à l'instant suivant, deux étapes sont exécutées dans l'ordre. En premier, dans l'étape de prédiction, l'état de chaque particule est mise à jour en utilisant le modèle d'évolution. En général, un modèle dynamique précis est nécessaire pour conserver les propriétés de robustesse de l'algorithme. En deuxième, dans l'étape de correction, l'information de mesure disponible est utilisée pour ajuster les poids des particules. Le poids de chaque particule est mis à jour en fonction de la vraisemblance de la particule pour décrire l'état actuel du système. Ce poids peut être calculé, via l'inférence bayésienne, pour qu'il soit proportionnel à la probabilité des mesures observées, en supposant que l'état actuel est identique à celui de la particule. Les états résultants sont re-échantillonnées à partir de la distribution de probabilité $p(x_k|Y_k)$ calculée dans le but d'utiliser des poids uniformes sur les particules échantillonnées pour une nouvelle itération de l'algorithme. A chaque instant, les caractéristiques du système (position, vitesse, etc) peuvent être calculées en utilisant les particules avec les poids associés pour une approximation de la fonction densité de probabilité. Pour plus de détails sur l'algorithme du filtrage particulaire et ses dérivées, le lecteur est invité à regarder [3, 6] et les références incluses.

3 Quelques concepts sur l'analyse par intervalles

Dans cette section, on présente brièvement l'analyse par intervalles via quelques définitions, et on fait le point sur quelques notions utiles pour comprendre les concepts de *propagation de contraintes* [7].

Un intervalle réel, noté $[x]$, est défini comme un sous ensemble fermé et convexe de \mathbb{R} , et un pavé $[\mathbf{x}]$ de \mathbb{R}^{n_x} comme un

produit cartésien de n_x intervalles: $[\mathbf{x}] = [x_1] \times [x_2] \cdots \times [x_n] = \times_{i=1}^{n_x} [x_i]$. Habituellement, l'analyse par intervalles est employée pour modéliser les quantités qui changent autour d'une valeur centrale et entre deux bornes, inférieure et supérieure, sans faire d'autre hypothèse sur la distribution entre ces bornes. Pour manipuler les intervalles, on doit introduire la fonctions d'inclusion $[f]$ de la fonction f , définie de manière à ce que l'image par $[f]$ d'un intervalle $[x]$ soit un intervalle $[f]([x])$ [7]. Plus précisément, cette fonction est calculée de sorte que l'intervalle contenant l'image soit optimal (le plus petit intervalle englobant l'image). Les opérations arithmétiques élémentaires telles que l'addition, la soustraction, ..., sont étendues pour être utilisées sur les intervalles. On peut aussi utiliser les opérations habituelles entre les ensembles de \mathbb{R}^n , i.e., $\subset, \supset, \cap, \dots$

Différents algorithmes, appelés *contracteurs*, ont été développés pour réduire la taille des pavés englobant la solution. Pour le problème de fusion considéré dans ce papier, on a choisi d'utiliser les techniques de propagation de contraintes [7], du fait d'une grande redondance de données ainsi que des équations.

Pour définir un *Problème de Satisfaction de Contraintes*, on considère un système de m relations f_m reliant des variables x_i d'un vecteur \mathbf{x} de \mathbb{R}^{n_x} par un ensemble d'équations de la forme: $f_j(x_1, \dots, x_{n_x}) = 0, j = 1 \dots m$, qui peut être écrit d'une manière compacte comme $f(\mathbf{x}) = \mathbf{0}$, où \mathbf{f} est le produit cartésien des f_j .

Définition 1 (problème de satisfaction de contraintes). *Un problème de satisfaction de contraintes \mathcal{H} est un problème qui rassemble un vecteur de variables \mathbf{x} appartenant à un domaine initial \mathbf{D} et un ensemble de contraintes \mathbf{f} reliant les variables x_i de \mathbf{x} .*

Notons que dans le cadre de l'analyse par intervalles, $\mathbf{D} = [\mathbf{x}] = \times_{i=1}^{n_x} [x_i]$. Le problème de satisfaction de contraintes consiste ainsi à trouver les valeurs de \mathbf{x} qui satisfont $f(\mathbf{x}) = \mathbf{0}$. L'ensemble des solutions est défini par $\mathbf{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}\{\mathbf{x}\} = \mathbf{0}\}$. Notons que \mathbf{S} n'est pas nécessairement un pavé. Dans le domaine de l'analyse par intervalles, résoudre le problème de satisfaction de contraintes consiste à *trouver le plus petit pavé* $[\mathbf{x}'] \subset [\mathbf{x}]$ de sorte que $\mathbf{S} \subset [\mathbf{x}']$.

Définition 2 (Contracteur). *Un contracteur est défini comme étant un opérateur utilisé pour contracter le domaine initial du problème de satisfaction de contraintes, et ainsi offrir un nouveau pavé $[\mathbf{x}'] \subset [\mathbf{x}]$ vérifiant $\mathbf{S} \subset [\mathbf{x}']$.*

Il existe plusieurs méthodes pour développer des contracteurs. Chacune de ces méthodes peut être adaptée à un problème spécifique de satisfaction de contraintes et pas un autre. La méthode utilisée dans ce papier est l'algorithme de Waltz basé sur la décomposition des contraintes en contraintes primitives et sur l'utilisation de techniques de propagation et de rétropropagation [7] sur les contraintes résultantes. Notons qu'une contrainte primitive contient seulement un opérateur arithmétique ou bien une fonction usuelle (cos, exp, etc.). Le principe du contracteur de Waltz est d'utiliser la propagation et la rétropropagation pour chaque contrainte, sans aucun ordre a priori, jusqu'à ce que le contracteur devienne inefficace. L'utilisation de ce contracteur peut être particulièrement efficace lorsqu'il y a une redondance de données

et d'équations. Notons que l'algorithme de Waltz est indépendant des non-linéarités et offre un contracteur localement consistant [7]. Pour plus de détails sur l'analyse par intervalles, le lecteur est invité à regarder [7] et les références incluses.

4 Algorithme hybride

Dans cette partie, la démarche d'aboutissement à l'algorithme hybride de filtrage particulière par pavés (appelé BPF comme Box Particle Filter) est présentée.

Dans une première étape, afin d'explorer l'espace d'état, on peut modéliser la région initiale de recherche par une union de N pavés $\{[x^{(i)}]\}_{i=1}^N$ disjoints et ayant le même poids. Après cette étape d'initialisation, l'état de chaque particule est mis à jour selon le modèle d'évolution en utilisant les outils d'analyse par intervalles. Ainsi, connaissant le nuage de pavés $\{[x^{(i)}]\}_{i=1}^N$ et les entrées $\{[u_{(k)}]\}$ à l'étape k , les pavés à l'instant $k+1$ sont construits en utilisant l'équation de propagation suivante: $[x_{k+1}^i] = [f]([x_k^i], [u_k])$, avec $[f]$ une fonction d'inclusion de f . On pourra remarquer que l'utilisation de l'approche à erreurs bornées dispense de l'introduction de bruits dans le modèle de propagation.

A ce stade, les nouvelles mesures à l'instant $k+1$ sont utilisées pour ajuster les poids des pavés ainsi que pour contracter leur volume. Pour cela, l'innovation apportée par la mesure doit être modélisée par une quantité devant indiquer la proximité entre la vraie mesure et celle prédite. Avec les approches à erreurs bornées, cette quantité peut être modélisée par une intersection entre le pavé de mesure et la prédiction de cette dernière. Ainsi pour chaque pavé $[x^{(i)}]$, $i = 1 \dots N$, une mesure est prédite en utilisant $[z_{k+1}^i] = [g]([x_k^i])$, où $[g]$ est une fonction d'inclusion de g . L'innovation $[r_{k+1}^i]$ peut être alors modélisée par l'intersection avec la boîte de mesure $[y_{k+1}]$ i.e. $[r_{k+1}^i] = [z_{k+1}^i] \cap [y_{k+1}]$. De ces innovations $[r_{k+1}^i]$, on peut déduire la vraisemblance de chaque pavé. On suit alors la règle suivante: un pavé pour lequel la mesure prédite n'a pas d'intersection avec la mesure réelle doit être pénalisé et un pavé pour lequel la mesure prédite est incluse dans la mesure réelle doit être favorisé. Cela nous conduit à modéliser la vraisemblance de chaque pavé selon la formule

$A^i = \prod_{j=1}^p A^i(j)$ où $A^i(j) = \frac{|[r_{k+1}^i(j)]|}{|[z_{k+1}^i(j)]|}$ et p est la dimension de la mesure et $|[X]|$ est la longueur de $[X]$. Remarquons que dans l'algorithme du filtrage particulière, chaque particule est propagée sans aucune information sur la variance de chaque particule (le poids de chaque particule donne uniquement une information sur la certitude du positionnement). A contrario, après l'étape de propagation, le volume de chaque pavé permet de prendre en compte l'imprécision découlant des erreurs de modèle et d'entrée. Dans l'optique de conserver un volume réduit pour chaque pavé, une nouvelle étape propre au BPF apparaît intéressante ici. Il s'agit d'avoir recours à un algorithme de satisfaction de contraintes afin d'éliminer les parties inconsistantes de chaque pavé [5]. Ce processus de contraction peut être vue comme une analogie à un volet de l'étape de correction du filtre de Kalman lorsque la matrice de variance-covariance est corrigée en utilisant la mesure [6]. On retrouve également cette étape de correction de matrice de variance-covariance dans le filtrage bayésien par somme de gaussiennes [1]. Par la suite, si l'innovation $[r_{k+1}^i]$ est non vide, on peut contracter

le pavé $[x_{k+1}^i]$ en utilisant le pavé intersection avec la mesure $[r_{k+1}^i]$ et l'algorithme de Waltz pour obtenir un nouveau pavé $[x_{k+1}^i]^{new}$. Dans le cas contraire $[x_{k+1}^i]^{new} = [x_{k+1}^i]$ (le pavé reste inchangé mais avec une vraisemblance nulle). Pour mettre à jour les poids des différents pavés on effectue une simple multiplication de la vraisemblance des pavés par leur précédent poids: $\omega_{k+1}^i = (\prod_{j=1}^p A^i(j))\omega_k^i = A^i\omega_k^i$. Puis, afin d'obtenir une somme des poids égale à 1, les nouveaux poids des pavés sont normalisés en utilisant la formule $\omega_{k+1}^i \leftarrow \frac{\omega_{k+1}^i}{\sum_{j=1}^N \omega_{k+1}^j}$. L'obtention de l'état estimé se fait, de façon naturelle en calculant la moyenne des centres des pavés selon leurs poids i.e. $\hat{x}_k = \sum_{i=1}^N \omega_k^i C_k^i$, où C_k^i est le centre du pavé i . En remarquant que l'estimé \hat{x}_k est calculée en utilisant N vecteurs C_k^i , une confiance sur l'estimation peut être obtenue en se basant sur la confiance de chaque C_k^i selon la formule $\hat{P}_k = \sum_{i=1}^N \omega_k^i P_k^i$, où P_k^i correspond à la confiance de chaque particule en utilisant son centre. En pratique, P_k^i peut être pris en prenant pour chaque dimension du pavé la moitié de la longueur de l'intervalle correspondant i.e. pour une dimension l , $\hat{P}_k(l) = \sum_{i=1}^N \omega_k^i \frac{|[x_k^i(l)]|}{2}$.

Après quelques itérations, apparaît le problème de la dégénérescence du poids des particules. La solution est alors de rééchantillonner les pavés en fonction de leurs poids [6]. Un problème inhérent à cet échantillonnage est la dépendance entre les nouvelles particules puisqu'il est probable que les particules dérivent toutes d'un nombre réduit d'ancêtres. Dans le cas du filtrage particulière, pour pallier à cette dépendance, un bruit artificiel est rajouté à chaque particule. Cette étape évite ainsi de faire diverger le filtre particulière. Au regard des possibilités offertes par l'utilisation des pavés, d'autres stratégies de rééchantillonnage peuvent être considérées [2]. Par exemple, afin d'obtenir des pavés indépendants et de taille réduite, il apparaît naturel de bissecter les pavés en fonction du nombre de leur réalisation après tirage puis d'accorder le même poids à toutes les pavés résultants.

5 Application à la localisation dynamique

On présente des résultats sur l'application de l'algorithme à la localisation d'un véhicule. Ces résultats montrent l'efficacité du BPF avec un nombre réduit de pavés en comparaison du filtrage particulière. A l'instant k , on note respectivement $\delta_{S,k}$ et $\delta_{\theta,k}$ les déplacements et rotations élémentaires donnés par les capteurs. Des tests spécifiques, en statique, ont été faits pour déterminer les bornes d'erreur sur les capteurs. La position et le cap du véhicule à l'instant k , $[X_k] = [x_k] \times [y_k] \times [\theta_k]$, sont calculés en utilisant le modèle suivant:

$$\begin{cases} x_{k+1} = x_k + \delta_{S,k} \cos(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ y_{k+1} = y_k + \delta_{S,k} \sin(\theta_k + \frac{\delta_{\theta,k}}{2}) \\ \theta_{k+1} = \theta_k + \delta_{\theta,k} \end{cases} \quad (2)$$

La mesure de la position est donnée par un système de positionnement global (GPS) (x_{GPS}, y_{GPS}) . Le point *longitude*, *latitude* estimé du GPS est converti dans un repère local et la borne d'erreur du positionnement est obtenue grâce à la trame *GST NMEA* [5]. La largeur de la mesure GPS peut alors être quantifiée en utilisant l'écart type σ_{GPS} sur x et y estimé

Table 1: Comparaison entre le PF et le BPF

	GPS	PF	BPF
EQM pour x(m)	0.134	0.129	0.119
EQM pour y(m)	0.374	0.217	0.242
nombre de particules	-	3000	10
temps de calcul d'une étape (ms)	-	666	149
EQM pour θ (degrés)	-	0.446	0.445

en temps réel par le récepteur GPS (trame GST), $[x_{GPS}] = [x_{GPS} - 3\sigma_{GPS}, x_{GPS} + 3\sigma_{GPS}]$, $[y_{GPS}] = [y_{GPS} - 3\sigma_{GPS}, y_{GPS} + 3\sigma_{GPS}]$. La mesure GPS ($[x_{GPS}], [y_{GPS}]$) est utilisée pour initialiser l'état ($[x_1], [y_1]$) à l'instant t_1 . N'ayant pas une mesure directe du cap, cette dernière va être initialisée par $[\theta_1] = [-\infty, +\infty]$.

Pour être capable de calculer une erreur d'estimation, nous avons eu recours à un récepteur GPS *Thales Navigation* utilisé en mode *Post-Processed Kinematic (PPK)* en travaillant avec une base locale (un Trimble 7400). On montre ci-après les résultats sur un trajet de 4.7 Km avec une vitesse moyenne de 50 Km/h, en utilisant un Pentium 4 à 3GHz et une implémentation Matlab.

La méthode de rééchantillonnage utilisée pour le BPF consiste en une stratégie de bisection. On échantillonne en premier les pavés suivant leurs poids en utilisant un algorithme classique de rééchantillonnage et, en deuxième, on divise chaque pavé tiré en plusieurs sous-pavés de nombre égal à la réalisation de la pavé initial. Ce type de rééchantillonnage permet de raffiner la solution autour des régions de grande vraisemblance et d'éliminer les pavés de poids faible. Cependant, on doit déterminer le nombre de divisions à faire pour chaque dimension. Pour l'état de trois variables considéré dans le modèle (2), le pavé est dans \mathbb{R}^3 . Nous suggérons de donner la priorité de bissecter l'angle $[\theta]$ des pavés du fait qu'on n'a pas une mesure directe sur cette variable, mais seulement une rotation élémentaire δ_θ du mobile. Cette division est adoptée tant que la taille de l'intervalle sur θ du pavé tiré est plus grande qu'une valeur fixée (deux degrés par exemple). Pour le choix entre la division des intervalles sur x ou sur y , on donne la priorité à l'intervalle de plus grande taille.

La Table 1 montre l'erreur quadratique moyenne (EQM) sur x et y pour le GPS, PF et BPF. Comme conclusion, le BPF et le PF donnent des performances équivalentes. Cependant, pour le BPF, nous avons utilisé 10 pavés seulement alors que pour le PF, 3000 particules ont été utilisées, ce qui fait un facteur de 300 entre les deux méthodes. La Table 1 donne en moyenne le temps d'exécution d'une étape pour chaque algorithme. Comme la fréquence de sortie de chaque filtre est de 5 HZ, on peut déduire que le temps d'exécution du BPF satisfait les contraintes temps-réel malgré l'utilisation de code Matlab pour faire les opérations arithmétiques des intervalles et sans aucune optimisation du code. Ceci n'est pas le cas pour le PF.

La Figure 1 montre les intervalles d'erreur pour x et y estimés pour le GPS (pointillés noirs), BPF (noir foncé) et PF (bleu). Pour le PF, l'intervalle d'erreur est calculée en utilisant une borne d'erreur de 3σ autour l'estimation. On peut remarquer que pour cette application les deux filtres sont consistants. Notons que l'intervalle d'erreur contient la valeur "0",

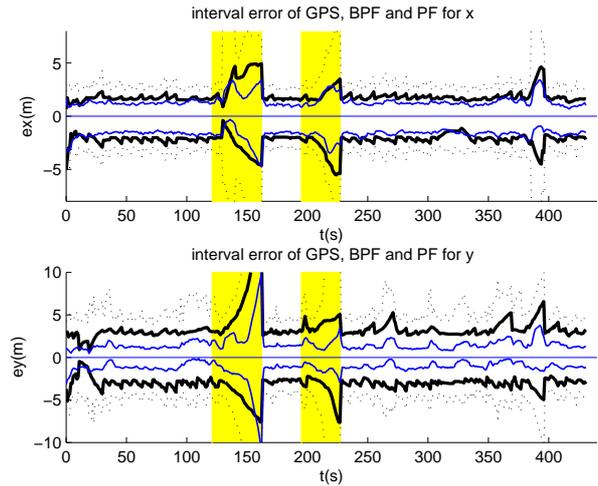


Figure 1: Intervalles d'erreur pour x et y estimés pour le GPS (pointillé noir), BPF (noir foncé) et PF (bleu continu).

ce qui veut dire que l'intervalle contient le point PPK.

6 Conclusion et perspectives

Un nouvel algorithme pour l'estimation d'état basé simultanément sur le filtrage particulaire et l'analyse par intervalle a été présenté. Les expérimentations sur des données réelles ont prouvé la faisabilité et l'efficacité de la méthode comparée au filtrage particulaire. De plus, le nouvel algorithme semble mieux adapté à des applications temps réels. Il semble particulièrement adapté lors de l'utilisation des cartes avec des routes rectangulaires. En effet, dans ce cas les pavés représentant l'état sont adaptées pour calculer une intersection avec la carte. Ainsi, les travaux futurs vont consister à appliquer le BPF sur des problèmes de localisation sur carte.

References

- [1] Alspach, D. L. and Sorenson, H. W.. Nonlinear Bayesian Estimation Using Gaussian Sum Approximations. *IEEE Transactions on Automatic Control* 17(4), 439-448, 1972.
- [2] Cendes, T. and Ratz, D. "Subdivision direction selection in interval methods for global optimization". *SIAM Journal of numerical Analysis*, 34:922-938, 1997.
- [3] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10:197-208, 2000.
- [4] D. Fox. adapting the Sample Size in Particle Filters Through KLD-Sampling. *The international Journal of robotics research*, vol. 22, pp. 985-1004, Dec 2003.
- [5] A. Gning and Ph. Bonnifait. "Dynamic Vehicle Localization using Constraints Propagation Techniques on Intervals. A comparison with Kalman Filtering". *ICRA 05*. Barcelona, April 2005.
- [6] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, vol. 50, pp. 425- 435, Feb. 2002.
- [7] Jaulin L., M. Kieffer, O. Didrit and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.